

Podstawy programowania 2023/24

Instrukcja projektowa

Gra Black Box

Autor: Andrzej Marczak

Wersja: 1.2

Black Box to gra planszowa dla jednego lub dwóch graczy, która symuluje strzelanie promieniami do czarnej skrzynki, aby wywnioskować lokalizacje "atomów" ukrytych w środku. Stworzył ją Eric Solomon. Gra planszowa została opublikowana przez Waddingtons w połowie lat 70. i przez Parker Brothers pod koniec lat 70.

Cel

Implementacja aplikacji konsolowej pozwalającej na rozgrywkę gry Black Box. Aplikacja powinna pozwalać na rozegranie kilku etapów o różnym stopniu trudności. Etapy różnią się wielkością planszy i liczbą ukrytych atomów. Plansze mogą mieć rozmiary 5×5, 8×8, 10×10. Liczba atomów może wynosić od 3 (dla najmniejszej planszy) do 8 dla największej planszy. Atomy ukryte na planszy powinny być rozmieszczone w sposób losowy. Dla każdego etapu na ekranie powinna być wyświetlana liczba ukrytych atomów. Strzały w kierunku atomów można wykonywać z pozycji każdej ściany. Każdy strzał powinien być numerowany. Tym samym numerem powinien być oznaczony wylot promienia. Oprócz tego powinno być oznaczone trafienie atomu (H) i odbicie promienia (R). Przypuszczalne położenie atomów oznaczamy małą literą o. Powinna być możliwa zmiana/usunięcie przypuszczalnego położenia atomów. Gra powinna umożliwić zaznaczenie przypuszczalnej pozycji atomów, równej liczbie ukrytych atomów dla danego etapu. Każdy etap jest oceniany pod względem liczby poprawnie zaznaczonych atomów. Jeśli gracz stwierdzi, że dalsza rozgrywka nie ma już sensu, może zrestartować aktualny etap. Jeśli gracz zaznaczy przypuszczalne położenie wszystkich atomów, powinien użyć klawisza k (koniec), wtedy program powinien wyświetlić położenie ukrytych atomów i liczbę. W przypadku poprawnego oznaczenia położenia atomu litera o zamienia się na O, w przypadku błędnego oznaczenia położenia, o zamienia się na X, a literą O oznaczone jest poprawne położenie atomu na planszy. Dodatkowo na ekranie powinna być wyświetlona liczba poprawnie zaznaczonych atomów. Z dowolnego etapu gry powinno być możliwe przejście do menu początkowego, w którym można:

- opuścić grę,
- rozpocząć nową grę od dowolnego etapu,

Oprócz tego gracz powinien mieć możliwość cofnięcia oraz ponowienia cofniętych wcześniej ruchów.

UWAGA: atomy są dla gracza NIEWIDOCZNE podczas rozgrywki!

Opis gry i przykładowe realizacje można znaleźć pod adresami:

[https://en.wikipedia.org/wiki/Black_Box_\(game\)](https://en.wikipedia.org/wiki/Black_Box_(game)) (opis na angielskojęzycznej Wikipedii)

<https://dosgames.com/game/the-black-box/#dosbox-div>

<https://www.chiark.greenend.org.uk/~sgtatham/puzzles/js/blackbox.html>

Środowisko

Program będzie implementowany z użyciem języka C++. Można używać dowolnych kompilatorów oraz edytorów tego języka. Używane narzędzia oraz IDE mogą współpracować z różnymi systemami operacyjnymi: Windows/Linux. **Proszę jednak pamiętać, że program oddawany będzie w laboratorium komputerowym na politechnicznym komputerze i tam powinien działać.** W programie można używać tylko standardowych bibliotek języka C++ za wyjątkiem Standardowej Biblioteki Szablonów (STL) i biblioteki C++ string.

Kryteria oceny

Zamieszczenie programu w wyznaczonym miejscu (na STOS-ie), kompilacja i uruchomienie programu są warunkiem przyznania punktów. Student musi sprawdzić czy w sali, w której będzie oddawał jest odpowiednie oprogramowanie. Projekt musi być wykonany samodzielnie przez oddającego studenta. Oddający musi być w stanie wytłumaczyć każdy element kodu oraz nanieść w nim drobne poprawki. Gra powinna być grywalna i wygodna w obsłudze. Oznacza to, że etapy początkowe powinny być możliwe do realizacji w możliwie krótkim czasie. Nie powinno być problemów w sterowaniu. Program nie powinien się zawieszać, lag'ować¹ oraz crash'ować² oraz wyrzucać wyjątki. Oprócz funkcjonalności programu oceniana będzie estetyka kodu oraz jego zgodność ze sztuką.

Obsługa programu

Program powinien wykorzystywać klawiaturę w następujący sposób:

- w, s, a, d i (W, S, A, D) – poruszanie się po planszy odpowiednio: w górę, w dół, w lewo i prawo;
- q, Q – wyjście do menu głównego;
- u, U – undo (cofnij ruch);
- r, R – redo (powtórz cofnięty ruch);
- spacja oddanie strzału (gdy kursor jest na dowolnej ścianie);
- o - na planszy umożliwia zaznaczenie przypuszczalnego położenia atomu;
- k – kończy rozgrywkę i umożliwia wyświetlenie rozwiązania i liczby uzyskanych punktów (poprawnie znalezionych atomów);
- p – umożliwia wyświetlenie rozwiązania (przerywa etap gry, brak możliwości kontynuowania tego etapu gry);
- H – Help/Pomoc – pokazuje na chwilę umieszczenie atomów na planszy
- pozostałe pozostają do wyboru przez programistę; Klawisze sterujące powinny być zatwierdzane klawiszem enter.

¹ [https://pl.wikipedia.org/wiki/Op%C3%B3%C5%BAnienie_\(informatyka\)](https://pl.wikipedia.org/wiki/Op%C3%B3%C5%BAnienie_(informatyka))

² [https://en.wikipedia.org/wiki/Crash_\(computing\)](https://en.wikipedia.org/wiki/Crash_(computing))

Wymagania i punktacja

Punkty z kolejnych etapów (wypunktowanie liczbami) naliczane są po spełnieniu wcześniejszych etapów. **Maksymalna liczba punktów za realizację pierwszego projektu wynosi 14.**

1. (3 pkt) Wyświetlanie ekranu powitalnego z nazwą programu z imieniem, nazwiskiem i numerem indeksu autora, mile widziana grafika w stylu ASCII art. Następnie wyświetlenie przynajmniej jednej planszy, o rozmiarach 5×5 z trzema ukrytymi atomami. Gracz może przemieszczać się kursorem po całej planszy i ścianach. W momencie zakończenia gry (klawisz "k") ukryte atomy mają zostać pokazane.
2. Menu
 - a. (1 pkt) Wyświetlanie planszy menu w dowolnym momencie rozgrywki, menu umożliwia, wyjście z gry i rozpoczęcie etapu od nowa,
 - b. (1 punkt) Opcja Help/Pomoc – pokazuje na chwilę umieszczenie atomów na planszy.
 - c. (1 pkt) Wiele etapów (z większą liczbą ukrytych atomów), możliwość wyboru etapu z poziomu menu.
 - d. (2 pkt) Opcja undo i redo (zagłębienie min 5 ruchów).
3. (1 pkt) Losowe rozmieszczenie atomów na planszy (przy każdym uruchomieniu etapu powinno być inne rozmieszczenie).
4. (1 pkt) Użycie znaków tabelki z rozszerzonych kodów ASCII do wyświetlenia obramowań planszy gry, plus estetyka programu
5. Plansze gry
 - a. (2 pkt) plansza 8×8 z 5 atomami.
 - b. (2 pkt) plansza 10×10 z 8 atomami.